

A novel image recovery method based on discrete cosine transform and matched blocks

Xingyuan Wang · Dandan Zhang · Xing Guo

Received: 27 December 2012 / Accepted: 16 April 2013 / Published online: 30 April 2013
© Springer Science+Business Media Dordrecht 2013

Abstract In this paper, we propose a method for image recovery based on Discrete Cosine Transform (DCT) and finding the best matched blocks by using a part of the fractal compression algorithm. At the same time, we propose a new check algorithm for checking if image blocks are tampered. First, the original image is divided into small blocks. The best matched block of each small block is searched in a particular way. Then the matching information is embedded as backup into other blocks. For the ones that fail to find the best matched blocks, DCT is applied on them and then quantized to be the backup. In order to prevent the backup of the tampered blocks from damaging, we generate 3 backups for each block and embed them into different quadrants. On the receiving side, the tampering check bits are extracted to localize the tampered areas, and the backup bits are used to restore the contents of the tampered regions. The experimental results have proved a good restorability of this algorithm, and the lower the tampering rate is, the better quality of restored content can be obtained.

Keywords Image recovery · Best matched block · Discrete cosine transform · Fractal · Check algorithm

1 Introduction

With the development of the Internet and multimedia, digital watermarking technology has received extensive business application as soon as it appeared. This technique is used for a wide range of applications, such as: copyright protection, source tracking (different recipients get differently watermarked content) and broadcast monitoring (television news often contains watermarked video from international agencies). It is a method to embed information into a multimedia element (such as: text, audio, image, video, or 3D models) without evident influence on the carrier. The needed properties of a digital watermark depend on the use case in which it is applied. According to the robustness, digital watermarking is typically classified into three categories: robust [1–5], fragile [6–9], and semifragile [10–12].

Robust watermarking is used for copyright protection and ownership verification. In this case, the embedded watermark bits should be robust against various attacks and can be extracted correctly. A digital watermark is called fragile if it fails to be extracted correctly and completely after the slightest modification. Detecting malicious tampering and recovering these locations is the most important characteristic of

X. Wang (✉) · D. Zhang · X. Guo
Faculty of Electronic Information & Electrical
Engineering, Dalian University of Technology, Dalian,
Liaoning 116024, P.R. China
e-mail: wangxy@dlut.edu.cn

D. Zhang
e-mail: zhangdandanllh@163.com

X. Guo
e-mail: guoxing_0425@126.com

fragile watermarking techniques. In fragile digital watermarking, the extraction algorithm should fail if any change is made to the signal. Semifragile watermarking technique is also for integrality authentication just like fragile watermark, but there is obvious difference between them. Semifragile watermarking is only sensitive to the change of the content characteristics. It can provide an effective solution for both the authentication of image content and the detection of malicious attacks. However, fragile watermarking can detect any tampering to the media.

In 1999, Fridrich et al. [13] proposed the self-embedding fragile watermarking scheme, in which the test image is divided into blocks to apply DCT (Discrete Cosine Transform) transform. Then the DCT coefficients were used to generate the authentication information. These watermarks are embedded for verifying the integrity of image and the recovery of tampered region. From then on, people began to research watermark techniques with self-embedding or self-recovery methods.

Zhang et al. [14] proposes two self-embedding watermarking schemes called a reference sharing mechanism, in which the watermark is derived from the original principal content in different regions and shared by these regions for content restoration. In the first scheme, the watermark is derived from the original data in five most significant bit layers. The second scheme is the hierarchical self-embedding scheme in which the host content is decomposed into three levels. Qian and Feng [15] present a method based on DCT to reduce the embedding data for self-recovery. The main operation is to encode different types of blocks with varied number of bits, and use the inpainting technology to recover the blocks with few details. Bravo-Solorio and Nandi [16] have combined both a secure block-wise resilient to cropping and an iterative pixel-wise mechanism to improve the tampering localization and self-recovery capabilities. The pixel-wise method can be iteratively repeated, so that a different bit-plane is watermarked every iteration. The block-wise method is the improved version of Fridrich's scheme where the authentication of each block is not independent of the others. But most methods cannot work when the tamper region is very large. Here, we propose a novel method based on best-matched blocks [17, 18] to overcome this problem.

Some pixel-wise schemes [19–22] used to detect tamper can accurately locate the modified pixels when

the tampered area is not too extensive, while block-wise schemes [23–25] only detect blocks containing fake contents. Liu et al. [19] proposed a fragile watermark scheme based on pixel-value difference image between the host image and the chaotic matrix pattern. This method breaks completely the unique dependence relationship between LSB and the pixel, and has good performance to resist vector quantization and oracle attacks. Zhang and Wang [23] proposed a tampered detection based on hash function. The reference-bits determined by the host image and the check-bits derived from the hash of blocks are embedded into the entire image by using a lossless DE embedding technique. But generally speaking, the method based on pixel-wise needs more localization watermark bits to embed. And the detecting method based hash function needs massive calculation. We propose a new checking algorithm for detecting tamper location.

In this paper, we propose a method for image recovery based on finding the best matched blocks by using a part of fractal compression algorithm. At the same time, we propose a new check algorithm for checking if image blocks are tampered. First, the original image is divided into small blocks. The best matched block of every small block is searched in a particular way. Then the matching information is embedded as backup into other blocks. For the ones that fail to find the best matched blocks, DCT is applied on them and the DCT coefficients are quantized to be as the backup. In order to prevent the backup of the tampered blocks from damaging, we generate three backups for each block and embed them into different quadrants. If the image is tampered, we can extract the backups to recover it. The organization of the paper is as follows. Section 2 describes DCT, quantization, and fractal coding. The proposed check algorithm for localizing the tampered location is presented in Sect. 3. Section 4 explains the proposed image recovery scheme. Experimental results are given in Sect. 5. Section 6 offers the conclusions.

2 Related works

2.1 DCT and quantization

The DCT, particularly the DCT-II, is often used in signal and image processing, especially for lossy data compression, because it has a strong energy compaction property: Most of the signal information tends

to be concentrated in a few low-frequency components of the DCT. The two-dimensional DCT formula is presented in Eq. (1):

$$\begin{aligned}
 F(u, v) &= c(u)c(v) \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \\
 &\quad \times \cos\left(\frac{2x+1}{2N}u\pi\right) \cos\left(\frac{2y+1}{2N}v\pi\right), \tag{1}
 \end{aligned}$$

where $x, y, u, v = 0, 1, \dots, N - 1$,

$$c(u) = c(v) = \begin{cases} 1/\sqrt{2}; & u = 0, v = 0, \\ 1; & \text{others.} \end{cases}$$

The inverse transform formula is described as follows:

$$\begin{aligned}
 f(x, y) &= \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v)F(u, v) \\
 &\quad \times \cos\left(\frac{2x+1}{2N}u\pi\right) \cos\left(\frac{2y+1}{2N}v\pi\right), \tag{2}
 \end{aligned}$$

where $x, y, u, v = 0, 1, \dots, N - 1$,

$$c(u) = c(v) = \begin{cases} 1/\sqrt{2}; & u = 0, v = 0, \\ 1; & \text{others.} \end{cases}$$

In our proposed method, the blocks size of 8×8 are used with two-dimensional DCT, and Eq. (1) can be changed to Eq. (3):

$$\begin{aligned}
 F(u, v) &= \frac{1}{4}c(u)c(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \\
 &\quad \times \cos\left(\frac{2x+1}{16}u\pi\right) \cos\left(\frac{2y+1}{16}v\pi\right), \tag{3}
 \end{aligned}$$

where $x, y, u, v = 0, 1, \dots, 7$,

$$c(u) = c(v) = \begin{cases} 1/\sqrt{2}; & u = 0, v = 0, \\ 1; & \text{others,} \end{cases}$$

$f(x, y)$ is the gray value of a pixel and $F(u, v)$ is the DCT coefficient.

Table 1 Typical quantization table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
79	92	95	98	112	100	103	99

Quantization in image processing is a lossy compression technique achieved by compressing a range of values to a single quantum value. When the number of discrete symbols in a given stream is reduced, the stream becomes more compressible. Human vision is much more sensitive to small variations in brightness over large areas than to the strength of high-frequency brightness variations. Therefore, the magnitudes of the high-frequency components are stored with a lower accuracy than the low-frequency components. The quantization matrix is designed to provide more resolution to more perceivable frequency components over less perceivable components in addition to transforming as many components to 0, which can be encoded with greatest efficiency. A typical brightness quantization table is shown in Table 1.

2.2 Fractal coding

Fractal compression is a lossy compression method for digital image, based on fractals [26–30]. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Fractal algorithm converts these parts into mathematical data called “fractal codes,” which are used to recreate the encoded image. With fractal compression, encoding is extremely computationally expensive because of the search used to find the self-similarities. Decoding, however, is quite fast with a partitioned iterated function system (PIFS) [31, 32].

First, the original image I (the size is $M \times N$) is partitioned into non-overlapping $r \times r$ blocks called range blocks R_i ($1 \leq i \leq (M \times N)/(r \times r)$) and overlapping $d \times d$ blocks called domain blocks D_i . The size of domain blocks must be greater than the size of range blocks to assure the transformations are contractive. All the domain blocks can be obtained by a

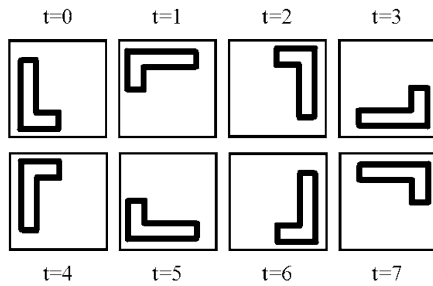


Fig. 1 8 kind of affine transforms

sliding window method, and the step length of domain blocks is δ . Then search a best-matched domain block for each range block R_i . We search the domain block pool to get a best-matched domain block D_i , an affine transform t and a contractive mapping w_i , which must assure that $w_i(D_i)$ has the minimum difference from R_i . There are 8 affine transforms (shown in Fig. 1), which contain identity ($t = 0$), 90° clockwise rotation ($t = 1$), 180° clockwise rotation ($t = 2$), 270° clockwise rotation ($t = 3$), x reflection ($t = 4$), y reflection ($t = 5$), $y = x$ reflection ($t = 6$), and $y = -x$ reflection ($t = 7$). The reconstruction process is a relatively simple iterative process.

3 Tampering check algorithm

Supposing a gray scale image block sized 8×8 (shown in Fig. 2), we will use 25 bits to be the checking bits. The proposed check algorithm is described as follows: view this image block as a cube of $8 \times 8 \times 8$ (8 bit-planes), and the top side is the presentation of the image content. X , Y and Z axes are shown in Fig. 2. The checking bits are defined as $c(0), c(1), c(2), \dots, c(24)$, $c(0), c(1), \dots, c(7)$ are used for recording the checking bits along Z axis, $c(8), c(9), \dots, c(15)$ are for that along X axis and $c(16), c(17), \dots, c(23)$ are for that along Y axis.

Partition the cube along Z axis into 8 layers and each layer contains $8 \times 8 = 64$ elements ('0' or '1'). Count the number of "1" in every layer and get 8 results ($Z(0), Z(1), Z(2), \dots, Z(7)$), ranging 0 to 64. If $Z(0)$ is an odd number, then $c(0) = 1$, else $c(0) = 0$; If $Z(1)$ is an odd number, then $c(1) = 1$, else $c(1) = 0$. And so on, for $c(2), \dots, c(7)$. Get the values of $c(8), c(9), \dots, c(15)$ along the X -axis and $c(16), c(17), \dots, c(23)$ along Y -axis in the same way.

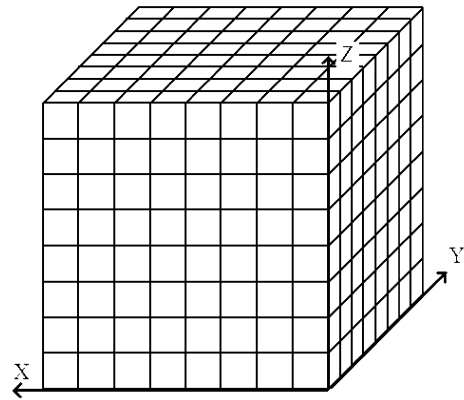


Fig. 2 Grayscale image block sized 8×8

The value of $c(24)$ depends on the parity of the number of "1" in the whole cube. If the number is odd, then $c(24) = 1$, else $c(24) = 0$.

4 Proposed watermarking procedure

A very important step in the fractal compression algorithm is that for each range block, the domain block pool is searched to find the best matched domain block which has minimum distortion with the current range block after transformation. We propose a new method which uses this step of fractal compression, finding the best matched domain block and storing the matching information as backups into other blocks.

In the fractal algorithm, suppose a cover gray scale image of 512×512 to be compressed, block D is the best matched block of block R . Now the storage information is:

- x , the X coordinate of D , which takes 8 bits;
- y , the Y coordinate of D , which takes 8 bits;
- t , the matching mode of D and R , which takes 3 bits;
- s , the pixel-value difference between D and R , which takes 9 bits.

We can see that, if block R and D are very similar and D is not damaged, the information of D can be used to recover the tampered block R . That is to say, D can be a backup of R . At the same time, the length of backup is relatively short. The length of pixels in R is $8 \times 8 \times 8 = 512$ bits, while the length of backup is just 28 bits.

4.1 Watermarking generation and embedding procedure

We propose a method for image recovery based on the above idea. First partition the image into smaller blocks, and search the image to find the best matched block for every block. Then embed the matching information (28 bits) as backup into other blocks. For the ones without the best matched blocks, DCT is applied on them and then quantized. The front 28 bits are extracted to be as the backup. In order to prevent the backup of the tampered blocks from damaging, we generate 3 backups for each block and embed them into different ones. If the image is tampered, we can extract the backups to recover it. The steps of watermark generation process are described as follows:

Step 1. Supposing a cover image sized 512×512 , partition it into 4 quadrants and smaller blocks sized 8×8 .

Step 2. For every block R in each quadrant, search a best matched block D in other three quadrants, respectively. D is determined by a threshold Δ . The matching information is described as follows:

- x , the X coordinate of D , 8 bits,
- y , the Y coordinate of D , 8 bits,
- t , the matching mode of D and R , 3 bits,
- s , the pixel-value difference between D and R , 9 bits,
- v , the effective bits, 2 bits.

So, $8 + 8 + 3 + 9 + 2 = 30$, $30 \times 3 + 25 = 115$, 25 bits are for tampering check, 30×3 represents 3 backups whose length is 30 bits. When $v = 0$, the content of backup is DCT coefficients. Otherwise ($v = 1, 2$ or 3), the content of backup is the information of the best matched block. The value of v means the level of similarity. When $v = 1$, the backup is the best matching information among the three available backups.

Step 3. For the ones that fail to find the best matched blocks, DCT is applied on them and then quantized. The front 28 bits (shown in Table 2) are extracted and add the effective bits v ($v = 0$) to be as the backup.

Step 4. Embed the matching information or quantized DCT coefficients into the bitplane 1 and bitplane 0 (LSB) of the cover image, to produce the watermarked image.

As shown in Fig. 3, three best-matched blocks of R in other three quadrants are searched. R is the block of 8×8 whose coordinates are $(32, 16)$ in the second quadrant. The size of the original image is 512×512 .

Table 2 Length assigning of quantized DCT coefficients

8	7	0	0	0	0	0	0
7	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

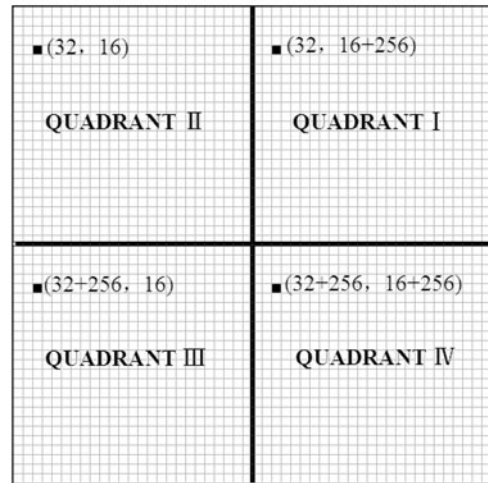


Fig. 3 Block R $(32,16)$ and its corresponding blocks

Search the best matched block in the first quadrant for R and record the matching bits as the backup, setting $v = 1$. If the best matched block can not be found within the threshold Δ , set the effective bits as 0 ($v = 0$), and record the quantized DCT coefficients as backup. Search the best matched block or quantize the DCT coefficients to be the matching backup in the third and fourth quadrants, respectively. The effective bits are set just as above.

For the backups with the effective bits $v = 1$, sort them based on the errors from small to large, and set the value of v following the description before. The matching bits in the first quadrant of R are embedded into the corresponding position block in the first quadrant, that is $(32, 16 + 256)$. The matching bits in the third quadrant of R are embedded into the corresponding position block in the third quadrant, that is $(32 + 256, 16)$. The matching bits in the fourth quadrant of R are embedded into the corresponding position block in the fourth quadrant, that is $(32 + 256,$

16 + 256). The bits in the bitplane 1 and bitplane 0 of watermarked R are shown Fig. 4 and Table 3.

4.2 Tamper detection and content restoration

On the receiving side, the tampering check bits are extracted to localize the tampered areas, and the backup bits are used to restore the contents of the tampered regions.

Step 1. Partition the tampered image into 4 quadrants and smaller blocks of 8×8 .

Step 2. Localize the tampered regions. Compare the tampering check bits extracted from each block with

the computed check bits from this block based on our proposed check algorithm. If they are exactly equal with each other, this block is not tampered, otherwise, it is damaged. According to the above description, we can get the tampered location.

Step 3. Based on the tampered blocks, detect the corresponding blocks in other three quadrants if they have been tampered. If not, extract the backup bits and recover the damaged blocks with the best matched information based on the effective bits v .

5 Experiment results

When using the proposed scheme to embed watermark into a test image sized 512×512 , the value of PSNR due to watermark embedded is about 44 dB (shown in Fig. 5), which is imperceptible and reasonable. Because in the watermark embedding procedure, the bitplane 1 and bitplane 0 of the cover image are replaced with the backup bits and checking bits. The size of small blocks is 8×8 , the step length of domain blocks is $\delta = 4$ and the threshold $\Delta = 3600$.

Table 3 Bits in the bitplane 1 and bitplane 0 of watermarked R

Shape	Length	Representation
○	25	Checking bits
△	30	Backup of block (32, 16 + 256)
□	30	Backup of block (32 + 256, 16)
☆	30	Backup of block (32 + 256, 16 + 256)

Fig. 4 Bits in the bitplane 1 and bitplane 0 of watermarked R

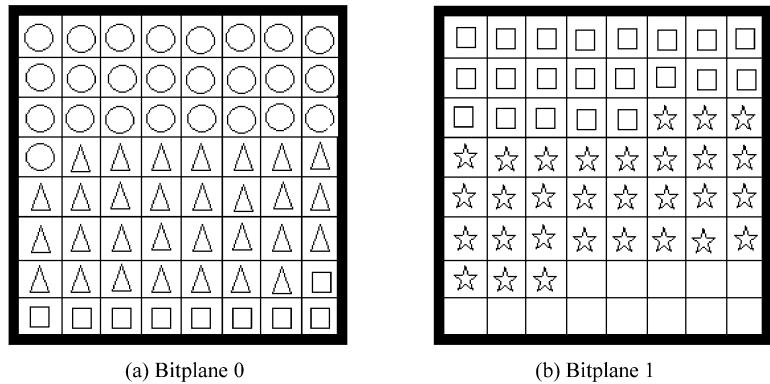
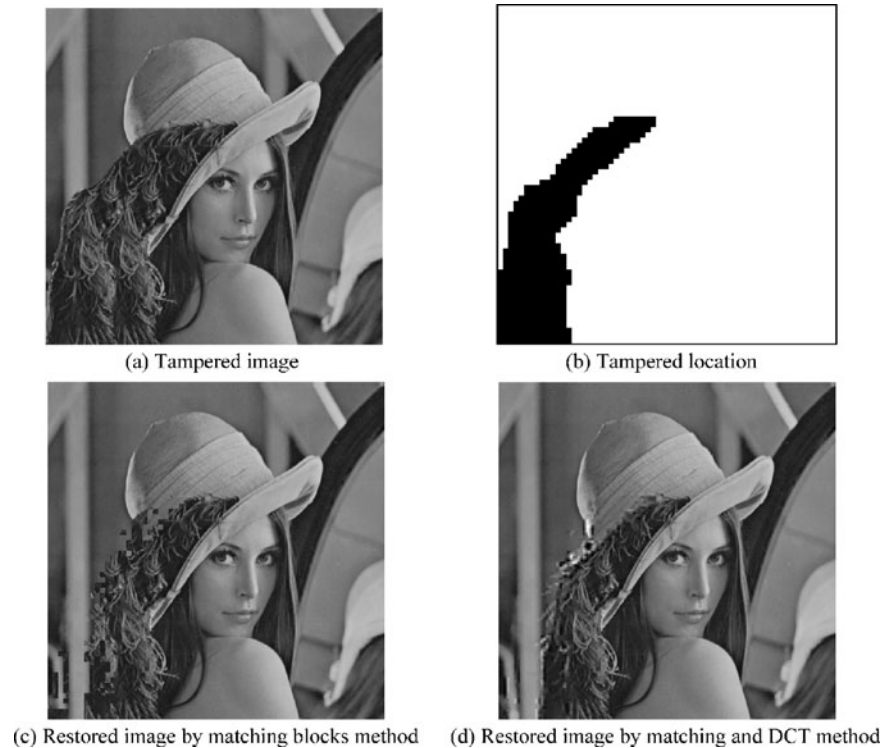


Fig. 5 Watermarked images

Fig. 6 Tampered location and recovery test

In Fig. 6, the proportion of tampered area is about 12 %. The PSNR of the restored image with matching blocks method is 30.3 dB where 216 blocks do not find the best matched blocks to recover the tampered areas. While the PSNR of the restored image with matching blocks method based on DCT is 33.3 dB and all the damaged blocks are recovered with backup bits and DCT coefficients. From Fig. 6(c), it is easy to find that the restored image with the matching blocks method has obvious blocking effect. But the result will be very good in the case where every block could find the best matched blocks. From Fig. 6(d), we can see for the blocks that could not find the best matched ones, applying DCT can be a better method to obtain good results.

Figure 7 shows the restored images recovered by our proposed method. Figure 7(a) is the first quadrant restored image, Fig. 7(b) is the second quadrant restored image, and Fig. 7(c) is the third quadrant restored image. Their values of the PSNR are 33.8 dB, 32.1 dB, and 29.0 dB, respectively. Figures 8 and 9 show the recovery effects of our proposed method when the tampering rates are 75 % and 23 %, respectively. Figure 8(a) is a damaged image whose first, second, and third quadrants are tampered. Figure 8(b) is

the tampered location and the percentage of the area is 75 %. Figure 8(c) is the recovered image, whose PSNR is 26.4 dB. Figure 9(a) is a tampered image whose top right corner is tampered. Figure 9(b) is the tampered location (23.4 %), and Fig. 9(c) is the recovered image (36.7 dB). From Figs. 7, 8, and 9, we can conclude that even though one or more of the four quadrants is tampered, the other quadrants can offer backups to recover the tampered areas.

Figure 10 gives the recovery performance of three tampered images Lena, Zelda, and Goldhill. Where the X -axis represents the tampering rate and the Y -axis is the PSNR of recovered images. It is evident that the quality of recovered image will decrease with the increase of tampering ratio. Even if the tampering rate is reach to 75 %, the recovered content has reasonable quality. Because the image is partitioned into four quadrants and for any block in one quadrant, it can find 3 backups in other quadrants.

In Fig. 11, we give the PSNR of the recovered image using our method and other methods in [7, 14, 20]. All the size of test images is 512×512 . The image in our algorithm and method in [20] is Lena and in [14] is the Couple. The value of PSNR in [7] is the average of all the test images. From this figure, we can observe



Fig. 7 Restored images

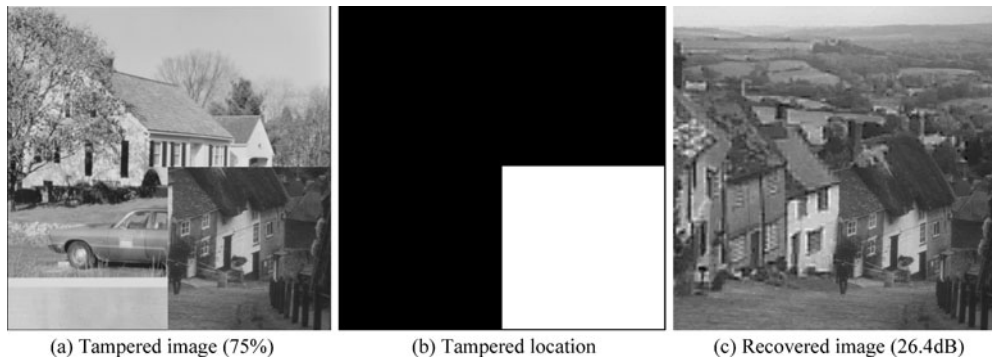


Fig. 8 Test images (75 %)

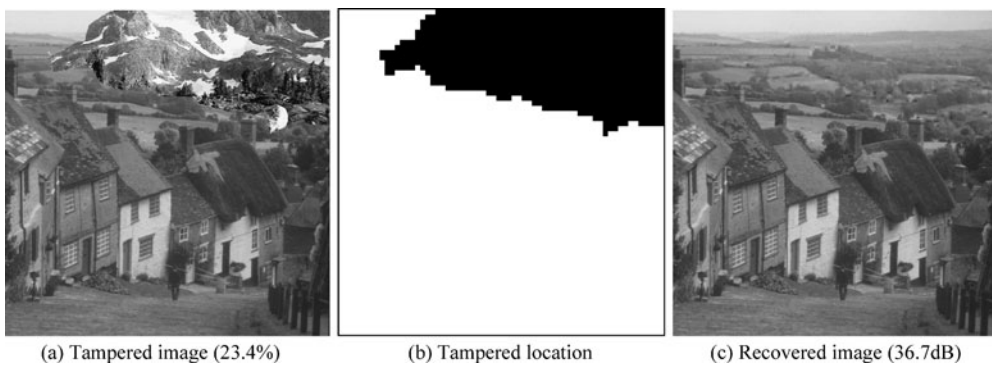


Fig. 9 Test images (23.4 %)

Table 4 Comparison of restoration capability among different schemes

Schemes	PSNR of the watermarked image	PSNR of recovered image	Condition of restoration
Method 1 in [14]	37.9 dB	$+\infty$	Tampering rate <24 %
Method 2 in [14]	37.9 dB	[22, 40] dB	Tampering rate <66 %
Method in [15]	37.9 dB	35.0 dB	Tampering rate <35 %
Method in [20]	37.9 dB	[22, 38] dB	Tampering rate <54 %
Method in [33]	51.1 dB	32.2 dB	Regions storing the original information of tampered areas must be reserved
Proposed method	44.3 dB	[26.4, 51.0] dB	Tampering rate <75 %

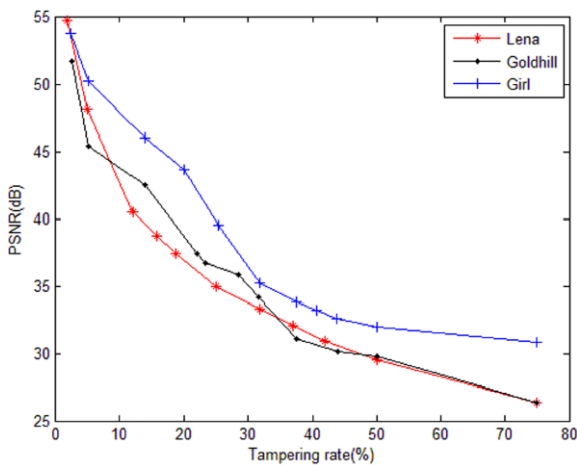


Fig. 10 Recovery performance of the tampered images

that our algorithm is better than other methods in the whole tampering rate region, and even the tampering rate is 75 %; the quality of the recovered image restored by our method is also acceptable.

Table 4 compares several fragile watermarking schemes with restoration capability. The method in [33] only works under the circumstances that regions storing the original information of tampered areas must be reserved and the first method in [14] does not work with a large tampering rate. By using the proposed scheme, the original content in an extensive area (75 %) can be recovered better than [15] and [20]. Also, the proposed method can get very ideal recovered image when the tampering rate is low.

6 Conclusion

In this paper, we propose a method for image recovery based on DCT and finding the best matched blocks by

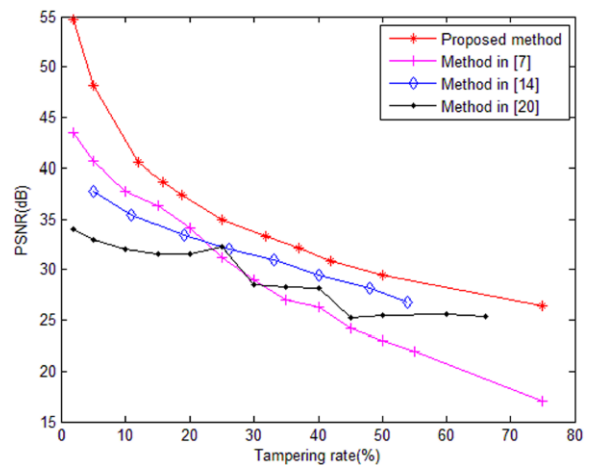


Fig. 11 PSNR of the recovered image using our method and other methods

using a part of fractal compression algorithm. Also, we propose a new check algorithm for detecting the tampered blocks. By partitioning the cover image into small blocks and searching the image to find the best matched block for every block, we view the matching information as the backup bits. For the blocks that fail to find the best matched blocks, DCT is applied on them and then quantized to be as the backup. In order to prevent the backup of the tampered blocks from damaging, we generate three backups for each block and embed them into different quadrants. The experimental results have proved a good restorability of this algorithm. And the lower the tampering rate is, the better quality of restored content can be obtained. That means the proposed scheme is more flexible than the previous methods.

Acknowledgements This research is supported by the National Natural Science Foundation of China (Nos. 61173183,

60973152, and 60573172), the Doctoral Program Foundation of Institution of Higher Education of China (No. 20070141014), Program for Liaoning Excellent Talents in University (No. LR2012003), the National Natural Science Foundation of Liaoning Province (No. 20082165), and the Fundamental Research Funds for the Central Universities (No. DUT12JB06).

References

- Xiang, S., Kim, H.J., Huang, J.: Invariant image watermarking based on statistical features in the low-frequency domain. *IEEE Trans. Circuits Syst. Video Technol.* **18**(6), 777–790 (2008)
- Farschi, S.M.R., Farschi, H.: A novel chaotic approach for information hiding in image. *Nonlinear Dyn.* **69**(4), 1525–1539 (2012)
- Run, R.S., Hong, S.J., Lai, J.L., Kao, T.W., Chen, R.J.: An improved SVD-based watermarking technique for copyright protection. *Expert Syst. Appl.* **39**(1), 673–689 (2012)
- Cancellaro, M., Battisti, F., Carli, M., Boato, G., Natale, F.G.B.D.: A commutative digital image watermarking and encryption method in the tree structured Haar transform domain. *Signal Process. Image Commun.* **26**(1), 1–12 (2011)
- Chang, C.Y., Wang, H.J., Su, S.J.: Copyright authentication for images with a full counter-propagation neural network. *Expert Syst. Appl.* **37**(12), 7639–7647 (2010)
- Hsieh, S.L., Tsai, I.J., Yeh, C.P., Chang, C.M.: An image authentication scheme based on digital watermarking and image secret sharing. *Multimed. Tools Appl.* **52**(2–3), 597–619 (2011)
- Li, C.L., Wang, Y.H., Ma, B., Zhang, Z.X.: A novel self-recovery fragile watermarking scheme based on dual-redundant-ring structure. *Comput. Electr. Eng.* **37**(6), 927–940 (2011)
- Lin, Z.C., He, J.F., Tang, X.O., Tang, C.K.: Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. *Pattern Recognit.* **42**(11), 2492–2501 (2009)
- Cao, Y.J., Gao, T.G., Fan, L., Yang, Q.T.: A robust detection algorithm for copy-move forgery in digital images. *Forensic Sci. Int.* **214**(1–3), 33–43 (2011)
- Qi, X.J., Xin, X.: A quantization-based semi-fragile watermarking scheme for image content authentication. *J. Vis. Commun. Image Represent.* **22**(2), 187–200 (2011)
- Peng, F., Guo, R.S., Li, C.T., Long, M.: A semi-fragile watermarking algorithm for authenticating 2D CAD engineering graphics based on log-polar transformation. *Comput. Aided Des.* **42**(12), 1207–1216 (2010)
- Zhu, X.Z., Ho, A.T.S., Marziliano, P.: A new semi-fragile image watermarking with robust tampering restoration using irregular sampling. *Signal Process. Image Commun.* **22**(5), 515–528 (2007)
- Fridrich, J., Goljan, M.: Images with self-correcting capabilities. In: 99th International Conference on Image Processing (ICIP), vol. 3, pp. 792–796 (1999)
- Zhang, X.P., Wang, S.Z., Qian, Z.X., Feng, G.R.: Reference sharing mechanism for watermark self-embedding. *IEEE Trans. Image Process.* **20**(2), 485–495 (2011)
- Qian, Z.X., Feng, G.R., Zhang, X.P., Wang, S.Z.: Image self-embedding with high-quality restoration capability. *Digit. Signal Process.* **21**(2), 278–286 (2011)
- Bravo-Solorio, S., Nandi, A.K.: Secure fragile watermarking method for image authentication with improved tampering localization and self-recovery capabilities. *Signal Process.* **91**(4), 728–739 (2011)
- Mirzaei, O., Yaghoobi, M., Irani, H.: A new image encryption method: sub-image encryption with hyper chaos. *Nonlinear Dyn.* **67**(1), 557–566 (2012)
- Wang, X.Y., Teng, L.: An image blocks encryption algorithm based on spatiotemporal chaos. *Nonlinear Dyn.* **67**(1), 365–371 (2012)
- Liu, S.H., Yao, H.X., Gao, W., Liu, Y.L.: An image fragile watermark scheme based on chaotic image pattern and pixel-pairs. *Appl. Math. Comput.* **185**(2), 869–882 (2007)
- Zhang, X.P., Wang, S.Z., Qian, Z.X., Feng, G.R.: Self-embedding watermark with flexible restoration quality. *Multimed. Tools Appl.* **54**(2), 385–395 (2011)
- Chang, C.C., Chen, K.N., Lee, C.F., Liu, L.J.: A secure fragile watermarking scheme based on chaos-and-hamming code. *J. Syst. Softw.* **84**(9), 1462–1470 (2011)
- Chen, W.C., Wang, M.S.: A fuzzy c-means clustering-based fragile watermarking scheme for image authentication. *Expert Syst. Appl.* **36**(2), 1300–1307 (2009)
- Zhang, X.P., Wang, S.Z.: Fragile watermarking with error-free restoration capability. *IEEE Trans. Multimed.* **10**(8), 1490–1499 (2008)
- Liu, Y.J., Zheng, Y.Q.: Adaptive robust fuzzy control for a class of uncertain chaotic systems. *Nonlinear Dyn.* **57**(3), 431–439 (2009)
- Liu, Y.J., Chen, C.L.P., Wen, G.X., Tong, S.C.: Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems. *IEEE Trans. Neural Netw.* **22**(7), 1162–1167 (2011)
- Hernandez-Avalos, P.A., Feregrino-Urbe, C., Cumplido, R.: Watermarking using similarities based on fractal codification. *Digit. Signal Process.* **22**(2), 324–336 (2012)
- Kiani, S., Moghaddam, M.E.: A multi-purpose digital image watermarking using fractal block coding. *J. Syst. Softw.* **84**(9), 1550–1562 (2011)
- Liu, Y.J., Wang, W., Tong, S.C., Liu, Y.S.: Robust adaptive tracking control for nonlinear systems based on bounds of fuzzy approximation parameters. *IEEE Trans. Syst. Man Cybern., Part A* **40**(1), 170–184 (2010)
- Liu, Y.J., Wang, W.: Adaptive fuzzy control for a class of uncertain nonaffine nonlinear systems. *Inf. Sci.* **177**(18), 3901–3917 (2007)
- Liu, Y.J., Tong, S.C., Chen, C.L.P.: Adaptive fuzzy control via observer design for uncertain nonlinear systems with unmodeled dynamics. *IEEE Trans. Fuzzy Syst.* **21**(2), 275–288 (2013)
- Zhang, T., Zhuang, Z.: Representation of discrete sequences with high-dimensional iterated function systems. *Nonlinear Dyn.* **48**(1–2), 49–57 (2007)
- Zhang, T., Liu, J.L., Zhuang, Z.: Representation of discrete sequences with N-dimensional iterated function systems in tensor form. *Nonlinear Dyn.* **52**(1–2), 89–93 (2008)
- He, H.J., Zhang, J.S., Chen, F.: Adjacent-block based statistical detection method for self-embedding watermarking techniques. *Signal Process.* **89**(8), 1557–1566 (2009)